PATENT ABSTRACTS OF JAPAN

(11)Publication number : **07-110768**
(43)Date of publication of application : **25.04.1995**

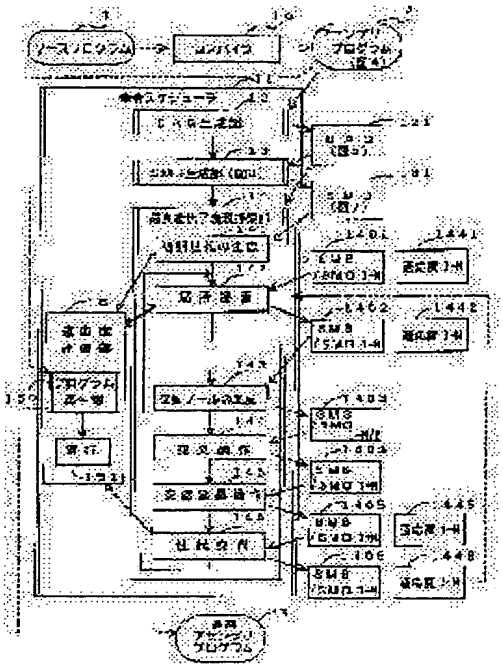
(51)Int.CI. G06F 9/38
G06F 9/45

(21)Application number : **05-255497**
(22)Date of filing : **13.10.1993**

(71)Applicant : **HITACHI LTD**
(72)Inventor : **UMETANI YUKIO**

(54) METHOD FOR SCHEDULING INSTRUCTION STRING

(57)Abstract:
PURPOSE: To provide a method for scheduling instruction strings usable in the RISC computers of a wide range in common.
CONSTITUTION: A DAG generation part 12 analyzes instruction dependence relation DAG (121) from an assembly program 2 and an SMD generation part 13 divides an original instruction string into plural strings respectively composed of a part of instructions to be successively executed and generates a string merging diagram(SMD) 131 for specifying the mutual dependence relation of the divided strings and the levels. An instruction scheduler 11 uses the DAG and SMD and searches the instruction string for which execution time is minimum or close to the minimum while rearranging the instruction strings corresponding to genetic algorithm.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平7－110768

(43)公開日　平成７年(1995)４月25日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ	9/38	３１０　Ｆ				
	9/45					
			9292－5Ｂ	Ｇ０６Ｆ　9/44	３２２　Ｆ	

審査請求　未請求　請求項の数30　ＯＬ　（全 20 頁）

(21)出願番号　特願平5－255497

(22)出願日　平成５年(1993)10月13日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　梅谷　征雄
東京都国分寺市東恋ケ窪１丁目280番地
株式会社日立製作所中央研究所内

(74)代理人　弁理士　小川　勝男

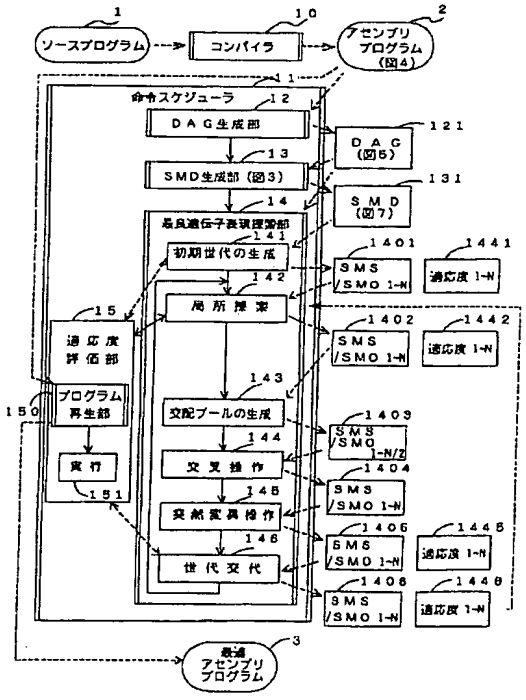
(54)【発明の名称】　命令列スケジュール方法

(57)【要約】　（修正有）

【目的】　広範囲のＲＩＳＣ計算機に共通して使える命令列のスケジュール方法を提供する。

【構成】　ＤＡＧ生成部１２がアセンブリプログラム２から命令依存関係ＤＡＧ（１２１）を解析し、ＳＭＤ生成部１３が、このＤＡＧから、元の命令列をそれぞれ順次実行されるべき一部の命令からなる複数のストリングに分割し、分割されたストリング相互の依存関係とそれらのレベルを指定するストリングマージ図（ＳＭＤ）１３１を生成する。命令スケジューラ１１はこのＤＡＧとＳＭＤとを使用して実行時間が最小あるいはそれに近い命令列を、遺伝的アルゴリズムに従って命令列を並べ替えながら探索する。

図１



1

【特許請求の範囲】

【請求項１】(a) プログラムを構成する命令列を解析して、それらの命令の間の依存関係を判別し、

(b) 遺伝的アルゴリズムに従って特定の計算機での該命令列の実行時間を少なくする、該判別された命令間の依存関係を満たす並び替え態様を探索し、その探索は、(b1) それぞれ該判別された命令間の依存関係を満たす命令並び替え態様を表す初期世代の複数の個体を生成し、(b2) 該複数の初期世代の個体を操作して、該特定の計算機での実行時間に関して、該複数の初期世代の個体のいずれか一つより改善された並び替え態様を表す複数の個体を含み、それぞれ該判別された命令間の依存関係を満たす命令並び替え態様を表す複数の次世代用の個体を生成し、(b3) 生成された複数の次世代用の個体を上記初期世代用の個体の代わりに使用してステップ(b2)(b3)を後続の複数の世代にわたって繰返し、(b4) 繰返し後に得られた複数の個体の内、該特定の計算機での実行時間に関して優れている一つの個体が表す並び替え態様を探索結果として選択するステップを有し、

(c) 該探索により得られた並び替え態様に従って該命令列を並び替えるステップを有する命令列スケジュール方法。

【請求項２】ステップ(b2)は、

該複数の初期世代の個体で表される並び替え態様により該命令列を並び替えたときに得られる命令列の該特定の計算機での実行時間を評価し、

該複数の初期世代の個体に特定の操作を施し、新たな命令の並び替え態様を表す複数の変形個体を生成し、

該複数の変形個体で表される並び替え態様により該命令列を並び替えたときに得られる命令列の該特定の計算機での実行時間を評価し、

該複数の初期世代の個体と該複数の変形個体から、該特定の計算機での実行時間が相対的に少ない複数の個体を該次世代用の個体として選択するステップを有する請求項１記載の命令列スケジュール方法。

【請求項３】該特定の操作を施すステップは、

該生成された初期世代の複数の個体を組み合せて複数の個体対を形成し、

それぞれの個体対に属する二つの個体を構成するデータの一部を交換して新たな一対の交叉変形個体を生成するステップを含む請求項２記載の命令列スケジュール方法。

【請求項４】該特定の操作を施すステップは、

該交換により生成された新たな複数の個体を構成するデータの一部をランダムに変更して、複数の突然変異個体を生成するステップをさらに含む請求項３記載の命令列スケジュール方法。

【請求項５】該特定の操作を施すステップは、

該複数の初期世代の個体の各々が表すデータの一部をその個体が生成された新たな複数の個体を構成するデータ

2

の一部をその個体が表す並び替え態様に近い並び替え態様を表すデータに変更することにより少なくとも一つの局所変形個体を生成し、

初期世代の各個体に対して生成された複数の局所変形個体で表される並び替え態様により該命令列を並び替えたときに得られる命令列の該特定の計算機での実行時間を評価し、

該複数の局所変形個体とその初期世代の個体に対して評価された実行時間に基づいて、該生成された複数の局所変形個体と初期世代の個体の一つを選択し、

上記生成、上記評価、上記選択を初期世代の他の複数の個体に対して繰り返し、もって複数の個体を選択し、

選択された複数の個体を、該複数の初期世代の個体の代わりに該複数の個体対の生成に使用するステップをさらに有する請求項４記載の命令列スケジュール方法。

【請求項６】該局所変形個体を生成するステップは、該複数の初期世代の個体の各々から複数の局所変形個体を生成するステップを有する請求項５記載の命令列スケジュール方法。

【請求項７】該複数の初期世代の個体に関する該特定の計算機での実行時間を評価するステップおよび該特定の操作で生成された複数の個体に関する該特定の計算機での実行時間を評価するステップは、

評価すべき個体が表す命令並び替え態様にしたがって、該命令列を並び替え、

並び替えにより得られた命令列を該特定の計算機で実行して、その命令列の実行時間を計測するステップを有する請求項２記載の命令列スケジュール方法。

【請求項８】初期世代の各個体は、各命令をマージすべきマージ先命令に関連する第１のデータと各命令のマージ順序に関連する第２のデータとを含む請求項１記載の命令列スケジュール方法。

【請求項９】初期世代の各個体は、上記プログラムの上記命令列を構成する、それぞれ相互に依存関係がある複数の命令からなる複数の部分命令列（ストリング）に対応して設けられた、それぞれのストリング内の複数の命令の位置として、それぞれの命令をマージすべき他の命令を示す複数の部分からなる第１のデータと、該複数のストリングのマージ順序を示す第２のデータを含む請求項１記載の命令列スケジュール方法。

【請求項１０】初期世代の各個体の該第１のデータの各ストリングに対応する部分は、そのストリングに属する複数の命令のマージ先の命令として、他の一つ又は複数のストリングに属する複数の命令の位置を指定するデータからなる請求項９記載の命令列スケジュール方法。

【請求項１１】複数の初期世代の個体の生成ステップ(b1)は、

各初期世代の個体に対して該第１のデータと該第２のデータをランダムに決定するステップを有し、

その決定ステップは、

各初期世代の個体に対して、該複数のストリングの各々内の各命令をマージすべ位置として、いずれかの他のストリング内の命令を、該命令間の依存関係で許容される命令群の中からランダムに選択し、
該選択結果に基づいて、該第１のデータを決定し、
該複数のストリングをマージすべきストリング順をランダムに決定し、
決定されたストリング順にしたがって、該第２のデータを決定するステップを有する請求項９記載の命令列スケジュール方法。
【請求項１２】該初期世代に操作を施すステップは、
該生成された初期世代の複数の個体を組み合せて複数の個体対を形成し、
それぞれの個体対に属する二つの個体を構成するデータの一部を交換して新たな複数の交差変形個体を生成するステップを含み、
各個体対の間でのデータの一部を交換するステップは、
その個体対の間で第１、第２のデータのいずれを交換するかをランダムに判定し、
その判定の結果、第１のデータを交換すると判定されたときには、該第１のデータの内、該複数のストリングの内のいずれに関する部分を交換するかをランダムに決定し、
その個体対の間でそれぞれの個体の第１のデータのうち、決定されたストリングに関する部分を交換し、
上記判定の結果、第２のデータを交換すると判定されたときには、その個体対の間でそれぞれの個体の第２のデータを交換するステップを有する請求項１１記載の命令列スケジュール方法。
【請求項１３】該初期世代に操作を施すステップは、
該データ交換により生成された複数の個体を構成するデータの一部をランダムに変更して、複数の突然変異個体を生成するステップをさらに含み、
該複数の突然変異個体を生成するステップは、
該データ交換により得られた複数の交叉変形個体の各々について、その第１、第２のデータのいずれを変更するかランダムに判定し、
その判定の結果、その個体の第１のデータを変更すると判定されたときには、その個体の第１のデータのうち、該複数のストリングの内のいずれに関する部分を変更するかをランダムに決定し、
その個体の該第１のデータのうち、該決定されたストリング内のいずれの命令に関する部分を変更するかをランダムに決定し、
その第１のデータのうち、その決定された命令のマージ位置を表す部分を、該命令間の依存関係により許容される範囲内でランダムに変更し、
上記判定の結果、第２のデータを変更すると判定されたときには、その個体の第２のデータをランダムに変更するステップを有する請求項１２記載の命令列スケジュール方法。
【請求項１４】該初期世代に操作を施すステップは、
該複数の初期世代の個体の各々が表すデータの一部をその個体が生成された新たな複数の個体を構成するデータをその個体が表す並び替え態様に近い並び替え態様を表すデータに変更することにより少なくとも一つの局所変形個体を生成し、
初期世代の各個体に対して生成された複数の局所変形個体で表される並び替え態様により該命令列を並び替えたときに得られる命令列の該特定の計算機での実行時間を評価し、
該複数の局所変形個体とその初期世代の個体に対して評価された実行時間に基づいて、該複数の複数の局所変形個体と初期世代の個体の一つを選択し、
上記生成、評価、選択を初期世代の他の個体に対して繰り返し、もって、複数の個体を選択し、
選択された複数の個体を、該複数の初期世代の個体の代わりに該複数の個体対の生成に使用するステップをさらに有し、
各初期世代個体に対する該局所変形個体の生成は、
該複数のストリングの少なくとも一つとそのストリング内の複数の命令の少なくとも一つを選択し、
その初期世代の個体の第１のデータの内、その選択された命令のマージ位置に関する部分を、その第１のデータが現に示すマージ位置の隣りの位置であって、上記命令間の依存関係により許容される位置を表すデータに変更した第１のデータを有する局所変形個体を生成するステップを有する請求項１２記載の命令列スケジュール方法。
【請求項１５】該局所変形個体の生成ステップは、各初期世代の個体に対して複数の局所変形個体を生成するステップを有し、そのステップは、
各初期世代の個体に対して、該複数のストリングがとり得るあらかじめ順序づけられた複数のストリング順のうち、その順から見て隣接する位置にあるストリング順を示すデータにその個体の第２のデータを変形した局所変形個体を生成するステップをさらに有する請求項１４記載の命令列スケジュール方法。
【請求項１６】判別された命令間の依存関係に基づいて、上記プログラムの上記命令列を、それぞれ相互に依存関係がある複数の命令からなる複数の部分命令列（ストリング）に、かつ、各命令がいずれか一つのストリングのみに含まれるように、分割し、
それぞれのストリングに含まれる複数の命令と他のストリングに含まれる複数の命令との間の上記判別された依存関係に基づいて、該複数のストリング間の依存関係を判別するステップをさらに有し、
初期世代の各個体の生成ステップは、該ストリング間の依存関係に基づいて、その個体を生成するステップを有する請求項１記載の命令列スケジュール方法。

【請求項１７】該複数のストリング間の依存関係に基づいて、各ストリングがストリング間依存関係のの階層内に位置するレベルを各ストリングに対して決定するステップをさらに有し、

該ストリング間の依存関係に基づいて、初期世代の各個体を生成するステップは、

各ストリング内の複数の命令をマージすべきマージ先ストリングとして、そのストリングが依存関係を有し、そのストリングよりレベルの低い一つ又は複数の他のストリングを選択し、

各ストリングに対して選択された一つ又は複数のマージ先ストリングに基づいて、その個体を生成する請求項１６記載の命令列スケジュール方法。

【請求項１８】各ストリングに対して選択された一つ又は複数のマージ先ストリングに基づいて、初期世代の各個体を生成するステップは、

各ストリング内の各命令のマージ位置として、そのストリングに対して上記選択された一つ又は複数のストリング内の、該命令間の依存関係によりマージが許容される範囲にあるいずれかの命令の位置を決定し、

該複数のストリングの複数の命令に対して決定されたマージ位置をあらわす第１のデータを含むデータをその個体をあらわすデータとして生成するステップを有する請求項１７記載の命令列スケジュール方法。

【請求項１９】初期世代の各個体を生成するステップは、

該複数のストリングのマージ順序を示す第２のデータを生成し、

該生成された第２のデータを、その個体に対して決定された第１のデータとの組み合せをその個体を表すデータと生成するる請求項１８記載の命令列スケジュール方法。

【請求項２０】初期世代の各個体を生成するステップは、

該複数のストリングをそれぞれのレベルの順に並べたストリング順を表すように第２のデータを生成するステップを更に有する請求項１９記載の命令列スケジュール方法。

【請求項２１】(a) プログラムを構成する命令列を解析して、それらの命令の間の依存関係を判別し、

(b) 判別された依存関係に基づいて、この命令列を、相互に依存関係がある複数の命令を含む複数の部分命令列（ストリング）に、かつ、各命令がいずれか一つのストリングのみに含まれるように、分割し、

(c) 上記ストリングをマージするためのストリング順と、それぞれのストリングの複数の命令をそれぞれマージするための、それぞれのストリング以外の他のストリング内のマージ位置とからそれぞれなる複数の並び替え態様を順次指定し、

(d) 各並び替え態様により指定された、ストリング順と

該複数のストリングの複数の命令のマージ位置とに基づいて、該複数のストリングを一つの命令列にマージし、

(e) 各並び替え態様に対して得られた命令列を特定の計算機で実行するに必要な実行時間を評価し、

(f) 該複数の並び替え態様に対して得られた複数の命令列の一つを、該複数の命令列に対して、上記ステップ(e)により得られた評価結果に基づいて、上記特定の計算機で実行すべき命令列として選択する命令列スケジュール方法。

【請求項２２】上記各ストリングの複数の命令のマージ位置を指定するステップは、それぞれの命令をマージする位置として、そのストリング毎に定めた他の一つ又は複数ののストリングのいずれか一つに属し、上記命令間依存関係を満たす位置を指定するステップを有する請求項２１記載の命令列スケジュール方法。

【請求項２３】上記ストリングのマージするステップは、

上記指定されたストリング順の先頭に位置するストリングに、その先頭以外の他の複数のストリングを、この指定されたマージ順に順次マージし、

この際、該複数の後続のストリングの各々のマージにあたっては、そのストリング内の各命令に対して指定されたマージ位置に従って、その命令をそのストリングに先行する他のストリングのマージの結果得られるストリングに順次マージして一つの命令列を生成するステップを有する請求項２１記載の命令列スケジュール方法。

【請求項２４】該複数のストリングの間の依存関係を、それぞれのストリングに含まれる複数の命令と他のストリングに含まれる複数の命令の間の上記判別された依存関係により判別し、

該複数のストリング間の依存関係に基づいて、各ストリングがストリング間依存関係のの階層内に位置するレベルを各ストリングに対して決定するステップをさらに有し、

上記各ストリングの各命令のマージ位置を指定するステップおよびそれを変更するステップは、そのストリングがそれに対して依存関係を有し、上記決定されたレベルがより低い他の先ストリングに属し、上記判別された命令間の依存関係を満たすマージ位置を指定するステップを有する請求項２１記載の命令列スケジュール方法。

【請求項２５】上記ストリング順のを指定するステップは、この判別されたストリングのレベル順に配列されたストリング順を選択するステップを有し、

上記ストリング順を変更するステップは、この判別されたレベル順に配列され、かつ、同一のレベルに属する複数のストリングの一部の順序が既に指定されたストリング順内の順序とは異なるストリング順を選択するステップを有する請求項２４記載の命令列スケジュール方法。

【請求項２６】上記各ストリングのレベルを決定するステップは、最長のストリングを最下層のレベルのストリ

ングとして選択するステップを有する請求項２４記載の命令列スケジュール方法。

【請求項２７】上記複数の列び替え態様を順次指定するステップは、上記特定の計算機の構造に依存しない手順で該複数の列び替え態様を順次指定するステップを有する請求項２１記載の命令列スケジュール方法。

【請求項２８】上記特定の計算機の構造に依存しない手順により該複数の列び替え態様を順次指定するステップは、

一つのストリング順と各ストリングの各命令のマージ位置を表すデータとの組からなる複数の遺伝子的表現を生成し、

生成された複数の遺伝子的表現を、遺伝的アルゴリズムに依存する手順で変更するステップを有する請求項２１記載の命令列スケジュール方法。

【請求項２９】上記複数の遺伝子的表現を変更するステップは、

該生成された複数の遺伝子表現のデータの一部を相互に交換して複数の新たな遺伝子表現を生成する交叉操作により生成し、

交叉操作により生成されたいずれかの遺伝子表現の少なくとも一部を突然変異操作による変更するステップを有する請求項２８記載の命令列スケジュール方法。

【請求項３０】上記複数の遺伝子的表現を変更するステップは、複数の遺伝子表現のデータの一部を、それぞれそのデータが表わす命令の並び替え態様に近い他の並び替え態様を表わす他のデータに変更する局所探索による生成を含む請求項２９記載の命令列スケジュール方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、計算機の性能を引き出すための命令列の並べ替え（スケジュール）の方法に関し、特にＲＩＳＣ(Reduced Instruction Set Computer)に好適な、命令列スケジュール方法に関する。

【０００２】

【従来の技術】半導体製造技術と回路設計技術の近年の進歩を反映したＲＩＳＣ計算機は複雑なパイプライン制御方式と命令起動方式を用いるため、命令列の実行時間はその並び順によって変化する。すなわち、命令列を許容範囲内で入れ替えてその性能を引き出すSequence Optimizerが重要な役割を果たす。性能を引き出すためには、依存関係を有する命令同志が隣接して並ばないように引き離すことが基本的に重要であるのみならず、並列に動作可能な演算器の構成と命令の組合せに適合した最適な入替えを行う必要がある。例えば、Phillip B. Gibbons,etc., "Efficient Instruction Scheduling fora Pipelined Architecture", Proc. SIGPLAN '86 Symposium on Compiler Construction,pp.11—16(以下公知文献[1]と呼ぶ)に代表されるように、従来の命令入替え方式では、依存命令の引き離しを目的とした幾つかの固定的な規則のみを用いて並べ替えを行っていた。

【０００３】

【発明が解決しようとする課題】上記の固定的な規則のみを用いた並べ替えではすべてのケースを尽くすわけではないので、十分な性能引出しがなされない場合がある。また計算機の論理構成や制御方式などの特性が変わる度に並び替え規則を変更する必要があった。本発明の目的は、スケジュールすべき命令を実行する計算機の論理構成あるいは制御方式に依存しない並び替え手順で、命令の並び替えできる命令列スケジュール方法を提供することである。

【０００４】

【課題を解決するための手段】この目的達成のために、本願第１の発明による命令列スケジュール方法は、(a)プログラムを構成する命令列を解析して、それらの命令の間の依存関係を判別し、(b)遺伝的アルゴリズムに従って特定の計算機での該命令列の実行時間を少なくする、該判別された命令間の依存関係を満たす並び替え態様を探索し、(c)該探索により得られた並び替え態様に従って該命令列を並び替えるステップを有する。

【０００５】本願第１の発明のより具体的な態様は、この命令間の依存関係に基づいて、上記プログラムの上記命令列を、それぞれ相互に依存関係がある複数の命令からなる複数の部分命令列（ストリング）に、かつ、各命令がいずれか一つのストリングのみに含まれるように、分割し、それぞれのストリングに含まれる複数の命令と他のストリングに含まれる複数の命令との間の上記判別された依存関係に基づいて、該複数のストリング間の依存関係を判別し、該命令間の依存関係と該ストリング間の依存関係とに基づいて、遺伝子アルゴリズムで使用する初期世代の各個体を生成するステップを有する。

【０００６】さらに、本願第２の発明による命令列スケジュール方法は、(a)プログラムを構成する命令列を解析して、それらの命令の間の依存関係を判別し、(b)判別された依存関係に基づいて、この命令列を、相互に依存関係がある複数の命令を含む複数の部分命令列（ストリング）に、かつ、各命令がいずれか一つのストリングのみに含まれるように、分割し、(c)上記ストリングをマージするためのストリング順と、それぞれのストリングの複数の命令をそれぞれマージするための、それぞれのストリング以外の他の一つ又は複数のストリング内のマージ位置とからそれぞれなる複数の並び替え態様を順次指定し、(d)各並び替え態様により指定された、ストリング順と該複数のストリングの複数の命令のマージ位置とに基づいて、該複数のストリングを一つの命令列にマージし、(e)各並び替え態様に対して得られた命令列を特定の計算機で実行するに必要な実行時間を評価し、(f)該複数の並び替え態様に対して得られた複数の命令列の一つを、該複数の命令列に対して、上記ステップ(e)により得られた評価結果に基づいて、上記特定の計

算機で実行すべき命令列として選択するステップを有する。

【０００７】

【作用】本願第１の発明では、特定の計算機での実行時間を少なくするように、命令の並び替えの態様を、遺伝的アルゴリズムにより探索するので、探索の過程においては、その特定の計算機の論理方式や制御方式に立ち入る必要があるので、計算機特性の変化に幅広く対応して命令列をスケジュールできる。さらに、本願第２の発明では、命令列の並び替え状態を、ストリングのマージ順と、各ストリング内の命令のマージ位置により指定するので、この指定を順次切り替えることにより、計算機での実行時間に関して最適あるいはそれに近い、命令列の並び替え態様を決定出来る。

【０００８】

【実施例】

（１）実施例の概要

図２に本発明による命令列スケジュール方法を実施するための計算機システムの全体構成を示す。このシステムは、命令列のスケジュールを実行するための計算機システム１０００と、スケジュールされる命令列を実行する、たとえばＲＩＳＣタイプの計算機２０００とからなる。前者は、中央処理装置（ＣＰＵ）１００、このＣＰＵの主記憶装置として使用されるメモリ２００、ファイル記憶装置３００、キーボードなどの入力装置４００、スケジュールの結果などを表示するためのＣＲＴ５００とからなる。ファイル記憶装置３００には、Ｃ言語等で書かれた、命令列のスケジュールを施すべきソースプログラム１を保持するとともに、それから後にコンパイルして得られるアセンブリプログラム２を保持するのに使用される。メモリ２００には、ソースプログラム１をアセンブリするための、それ自体公知のコンパイラ１０と、命令列のスケジュールをするためのプログラムである命令列スケジューラ１１とが格納される。

【０００９】本実施例では、このスケジューラ１１が、アセンブリプログラム２を解析して、そのプログラムの構造を表す情報として、このプログラムに含まれる命令間の依存関係を表す命令依存解析図（Dpendency analysys grapf－ＤＡＧ）１２１を得るとともに、このプログラムをそれぞれ複数の命令からなる複数のストリングに分割し、それぞれのストリングの相互の依存関係を表すストリングマージ図（String merge state daiagram－ＳＭＤ）１３１とを得る。これらの情報をベースにして、スケジューラ１１は、このプログラムの命令列の並び替えを予め定めた一定の並び替え手順で行なう。具体的には、本実施例では遺伝子アルゴリズムに従って行なう。この際、それぞれの命令列の並びを表す情報として、ストリング内の命令のマージ位置を表す情報であるストリングマージ状態（String mergestate－ＳＭＳ）およびストリングのマージの順序を表す情報であるストリングマージオーダ（String merge order－ＳＭＯ）との組み合せ１４０１、、、１４０７を使用するようになっている。各並び替えの結果については、プログラム再生部１５０によりその並びに対応したアセンブリプログラムを生成し、実計算機２０００（図２）で実際に実行して、その適応度としての実行時間を測定し、その並び替えの良否の評価に使用し、その後の並び替えに反映し、最終的には最適な並び替えを決定するようになっている。

【００１０】とくに、本実施例では、命令列の並び替え態様を表すデータを遺伝的アルゴリズムに従って、操作して、最適又はほぼ最適な命令列並び替え態様を決める所に特徴がある。より具体的には、プログラムを命令間の依存関係に基づいてストリングに分けて、各ストリング間の依存関係に基づいて各ストリングのマージ先ストリングを決め、命令間の依存関係の制約の下で許される範囲で、ストリングの各命令のマージ位置を変え、ストリング間の依存関係の制約下で許される範囲でストリングのマージ順序を変えることにより、命令列の並び替えを種々行ない、得られる種々の命令列の良し悪しは、その計算機の特性を考慮して判別する。これらの操作が遺伝的アルゴリズムに従って行なわれる。

【００１１】その他の特徴は以下で説明する。

【００１２】（２）遺伝的アルゴリズムによる命令列並び替え処理の概要

遺伝的アルゴリズム(Genetic Algorithm)それ自体は、公知である。例えば、D. E. Goldberg : "Genetic Algorithm in Search, Optimization and Machine Learning", Addison—Wesley, 1989, pp.1-23(以下参考文献[2]と呼ぶ）ないし H. Muehlenbein, D. Schlierkamp—Voosen : "Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimization", EvolutionaryComputation 1, MIT Press,1993,pp.1-18(以下参考文献[3]と呼ぶ）参照。遺伝的アルゴリズムは、生命体の進化・適応過程の数学的モデルであり、適当な遺伝子で表現される個体集団に対して、局所探索(後天的適応)、交配(交叉)、突然変異などの遺伝的操作を施し、その結果から新世代の個体を生成し、それに対して以上の処理を繰返し実行し、こうして、世代を追って集団としての改良を図って行くものである。従って、遺伝的アルゴリズムを命令の並び替えに適用すれば、原理的に命令列の許される全ての並べ替えを候補とすることができ、そこから効率良く最良のものを探索するので、従来方式に較べて良いものが選ばれる可能性が高い。また探索の過程で用いる適応度の評価尺度に計算機特性を反映しておけば自動的に尺度に応じた最良解が選択されるので、計算機特性の変化にも幅広く対応できる。

【００１３】遺伝的アルゴリズムを命令列の並べ替えに適用する上で考慮すべき点は、遺伝子として使用する"命令の並べ替え状態"の表現形式と遺伝的操作の定義で

あり、これらが探索の質と効率を大きく左右する。命令の並べ替え状態を表す素朴な表現は命令番号を出現順に並べた数値列(順列表現)である。これに通常の遺伝的操作である交叉や突然変異を行うと、これらの操作は、元の数値を部分的に入れ替えたり他の値に変更するだけであるので、これらの操作の結果得られる数値列は、複数の位置に同じ値をもつ数字列となり、もはや順列表現ではなくなる。従ってこの結果に対するその後の補正が必要である。本実施例ではこの点を考慮して命令の並びを表すデータとして遺伝的アルゴリズムに適したもの（以下、これを簡単化のために遺伝子表現という）を工夫した。

【００１４】なお、遺伝的アルゴリズムの応用にあたっては、一般的には、改良の程度を示す尺度となる適応度関数の決定も問題となるが、命令列の並び替えへの適用にあっては、適応度関数としては、並べ替えられた命令列を特定の計算機で実行したときの実行時間ないし実行サイクル数を使用する。従来から命令列を構成する命令間の依存関係を表現するために命令解析グラフ(Dependency Analysis Graph－ＤＡＧ)と呼ぶグラフが用いられており、命令列の並べ替えもＤＡＧの表現する依存関係の制限内で行われなければならない。本実施例でもこのＤＡＧは使用する。本実施例では、新たな概念としてストリングと呼ぶ命令間の依存関係でつながれた、元の命令の一部の命令からなる部分列を導入する。ストリングはＤＡＧ上の一次元の部分グラフであり、元の命令列は、ＤＡＧに依存して、複数のストリングに分解できる。各命令はいずれか一つのストリングに属し、複数のストリングに重複して属さない。この意味で、各ストリングは他のストリングに対して排他的である。ストリングは依存関係でつながれた糸状の命令の列であるので、それを続けて実行すると、その内の一つの命令が終わらないと次の命令を実行できない。この結果、ＲＩＳＣ計算機内のパイプラインに空きを生じ、実行性能が低下する。そこで複数のストリングをマージして実行することがスケジューリング上必要である。

【００１５】そのためストリング間のマージ状態を表現するコードとしてストリングマージ状態コード(String Merge Status－ＳＭＳ)を導入する。ＳＭＳコードの内容はストリング間のマージ構造(どのストリングをどのストリングにマージさせるか)に依存するが、できるかぎりマージの階層数を減らし、ストリングごとのコードの設定が独立にできるようにした方が表現の冗長性が少なく、遺伝的アルゴリズムによる探索の効率をあげることが期待できる。そのためＤＡＧで表現された命令間の依存関係で結ばれたストリングの間でのみマージ関係を作ることとした。具体的には、各ストリングのＳＭＳコードは、そのストリングの各命令をマージする、マージ先ストリング内の位置を表す。全ストリングのマージ構造を決めるために、すべてのストリングがそこにマージされる基底のストリングを一つ定めた後、そこからの逆マージ系列の最長の長さによりストリングにレベルを付け、ストリング全体を階層的に組織化する。このストリング間の階層構造をストリングマージ図(String Merge Diagram－ＳＭＤ)と名付ける。

【００１６】さらにこの階層構造の中で、複数のストリングが共通のマージ先ストリングを持ちかつマージ点の競合がある場合に優先度を決めるため、ストリングの優先度リストであるストリングマージ順序(String Merge Order－ＳＭＯ)とよぶ別のコードを導入する。ＳＭＯコードは、ストリングの名称をレベル順に並べたものである。ＤＡＧの制約内での命令列の並び替えは各ストリングを構成する命令の順番を変えずストリング同志のマージ(混合)状態のみを変えるので、命令列の並べ替え状態は、ストリングごとのＳＭＳコードとひとつのＳＭＯコードで表現できる。これを遺伝子表現に用いる。

【００１７】遺伝的アルゴリズムを実現するためにはさらに、、局所探索(Local Search)、交叉(CrossoverまたはRecombination)、突然変異(Mutation))等の遺伝的操作(Genetic Operation)を定義する必要がある。本実施例では上記の遺伝子表現を用いて、交叉操作を、２つの遺伝子表現の間での特定ストリングのＳＭＳコードの交換ないし特定レベル部分のＳＭＯコードの交換と定義する。同じく突然変異操作を、選択したストリング群のＳＭＳコードないし選択したレベル群のＳＭＯコードの現在値からの変更と定義する。局所探索操作はＳＭＳコードないしＳＭＯコードのうちの１ヶ所の、適応度の改善をもたらすような隣接値への変更と定義する。各ストリングの各命令のマージ位置を示すＳＭＳコードと、ストリングのマージの優先度を示すＳＭＯコードとの性質から、上記の遺伝的操作の結果は、自動的に後補正の必要がない。さらに、遺伝的アルゴリズムに従い命令列の並び替えを行なうので、特定の計算機の構造に依存しない手順で命令列の並び替えを行なうことが出来る。遺伝的アルゴリズムを実現するためにはまた、適応度の評価が必要であるが、本実施例では、並べ替えられた命令列の特定ＲＩＳＣ計算機での実行時間で図った実行サイクル数を適応度とする。当然、その値が小さいほど高い適応度を意味する。

【００１８】（３）スケジューラ１１の詳細

図１に従い、本実施例の処理の詳細を説明する。ソースプログラム１からコンパイラ１１により得られたアセンブリ語の命令列(アセンブリプログラム)２が得られる。図４にソースプログラム１をコンパイルして得られるアセンブリプログラム２の１例を示す。この表記では左から、命令番号、ラベル、命令コード、第１入力データを保持する特定のＲＩＳＣ計算機内のレジスタの番号、第２入力データないしそれを保持するレジスタ番号、出力データを保持するレジスタ番号を示す。６，８，１１，１８番命令のような主記憶装置からのロード／ストア命

令では、鍵カッコのなかはアドレスレジスタの番号を示す。

【００１９】アセンブリプログラム２は、命令スケジューラ１１により、以下のように処理される。

（３－１）ＤＡＧ生成部１２

これにより、このアセンブリプログラム２に対するＤＡＧ１２１が生成される。ＤＡＧ１２１の生成は、たとえば、Jeanne Ferrante "The Program Dependence Graph and Its Use in Optimization",ACM Transaction on Programming Language and Systems, Vol.9, No.3, July 1987, pp.319—349(以下参考文献[４]と呼ぶ)などに示される公知の方法で行われる。図４のアセンブリプログラムから生成されたＤＡＧ１２１を図５に示す。ここで丸括弧内の数字は命令番号を示す。線(アーク)は命令間の依存関係を示し、線の上の命令の結果を下の命令が使うか上下の少なくても一方が分岐命令であること、従って上の命令が下の命令に先立って行われるべきことを意味する。ＤＡＧ１２１の生成は通常のコンパイラで実施しているとおり、命令間のレジスタの参照関係や演算命令と分岐命令等の制御命令の順序関係を解析して行われる。

【００２０】（３－２）ＳＭＤ生成部１３

続いてＳＭＤ生成部１３がＤＡＧ１２１よりＳＭＤ(String Merge Diagram：ストリングマージ図)１３１を生成する。その手順を図３を参照して説明する。

（ステップ３１）まず、ＤＡＧ１２１をストリングに排他的に分解し、その結果としてストリングベクトル(String Vector－ＳＴＶ）３１１（図６）を得る。ストリングはＤＡＧで示される命令間の依存関係のアークで順次つながれた複数の命令からなる部分命令列である。命令列のストリングへの分解は以下の手順で行われる。

【００２１】まずＤＡＧ内で上からの依存関係を持たない任意の一つの命令(例えば、命令(１))選択し、そこを始点としてアークの一つを任意に選択しながら下方にアークが尽きるまで辿って第１のストリング(この場合は(Ａ))を抽出する。続いて未抽出の命令からの依存関係を持たない命令を任意に一つ選び、そこを始点として未抽出命令への下方アークが無くなるまでアークの一つを任意に選んで辿る操作を繰り返して、残りのストリング(この場合は(Ｂ),(Ｃ),(Ｄ),(Ｅ))を抽出する。例えば(Ｂ)については始点命令(７)からアークをたどって(８)を抽出するが次の（９）は既抽出であるのでここでおわりとなる。ストリングの抽出には任意性があるがそれで構わない。図５のＤＡＧに対して得られたＳＴＶの例を図６に示す。ＳＴＶは２次元のリスト構造で保持する。図６でアルファベットはストリングの名称、続く番号列はストリングを構成する命令列を示す。

【００２２】（ステップ３２）図３に戻り、このステップで、ストリングのレベルを決定する。各ストリングのレベルは、ストリング間のマージ状態を階層的に整理して管理するために導入するものであり、低レベルのストリングに高レベルのストリングがマージされるように、マージ時に使用する。レベル付けは命令間の依存関係から派生するストリング同志の依存関係に基づいておこなう。各々のストリングに属する命令の間にＤＡＧで規定される依存関係があるとき、ストリング同志は依存関係を持つという。たとえば図６でストリング(Ａ)の命令（９)はストリング(Ｂ)の命令(８)と図５のＤＡＧ上依存関係があるので(Ａ)と(Ｂ)は依存関係を持ち、(Ｂ)と(Ｃ)は持たない。各ストリングのレベルは、より低位のストリングのレベル値がより小さくなるように決める。

【００２３】具体的には、まず、最低位のストリングを選びそれにレベル１を与える。レベル付けは、最低位のストリングに何を選ぶかにより任意性があるが基本的に何であっても構わない。ここでは最も長いストリング(Ａ)を選びそれにレベル１を与える。その理由は以下のとおりである。後に説明するように、各ストリングにマージコード（ＳＭＳコード）を付与する。その際、最低位のストリングにマージコードが不要である。さらに、各ストリングのＳＭＳは、そのストリングの命令の数に応じて長くなる。処理の高速化のために、ストリングのＳＭＳコードの最大長ををを短くすることが望ましい。従って、最も長いストリング(Ａ)を選びそれにレベル１を与えることは、処理の高速化の上で望ましい。

【００２４】続いて最低位のレベルのストリング(Ａ)と依存関係を持たないストリングがあればそれにもレベル１を与える。図６の例では、そのようなストリングはない。続いてレベル１のストリングあるいはレベルの与えられていないストリングとは依存関係はあるが、それらのストリング以外のストリングとは依存関係のないストリングストリング、例えば、(Ｂ),(Ｃ),(Ｄ)にレベル２を与える。さらにレベル１，２と依存関係を有するが、それ以外のストリングとは依存関係を有しないストリング(Ｅ)にレベル３を与える。以下同様にこの手順をすべてのストリングにレベルが与えられるまで続ける。

【００２５】（ステップ３３）最後にステップ３３により、ストリングのレベル情報とＤＡＧ１２１にもとずいて、ストリングを階層化して図７に例示するような情報を有するストリングマージ図（ＳＭＤ）１３１を生成する。このＳＭＤ１３１は、少なくとも、各ストリングを構成する命令とそれらの実行順と、各ストリングのレベルとストリング間の依存関係を示す情報を有する。図７の矢印は依存関係を有するストリングの間で高レベルから低レベルに向かって付けられ、２つのストリングがマージ元ストリングとマージ先ストリングの関係にあることを明らかにする。このストリングの依存関係は、以上のレベルの決め方から明かであるように、ＤＡＧ１２１による命令の依存関係に依存して決められている。なお、図６の例では、ストリング(Ｂ)(Ｃ)(Ｄ)はストリング(Ａ)にのみ依存するが、ストリング(Ｅ)は、ストリン

グ(Ｂ)に依存するだけでなく、ストリング(Ａ)にも依存する。こうしてＳＭＤ生成部１３の処理が終了する。

【００２６】（３－３）命令列の並び替え状態を表すデータ

図１に戻り、最良遺伝子表現探索部１４は、生成されたＤＡＧ１２１とＳＭＤ１３１を用いて、命令列の並びを表すデータ（遺伝子表現）を生成し、さらに、遺伝的アルゴリズムに従ってその遺伝子表現を順次更新し、それにより命令列最良遺伝子表現の探索を行う。最良遺伝子表現探索部１４の詳細説明の前に、本実施例で使用する命令列の並びを表す遺伝子表現を先に説明する。

【００２７】本実施例では、命令の並び替えは、プログラムを構成する複数のストリングの各々の命令を他のストリングにマージする位置を変えて、かつ、ストリングのマージ順をかえて行なう。このような命令の並び替えを行なうために、本実施例では、全てのストリングに対するストリングマージ順序コード（ＳＭＯコード）と、全ストリングに対する一群のストリングマージ状態コード（ＳＭＳコード）との組を使用し、かつ、この組を遺伝的アルゴリズムにより命令列を並び替えるために、命令列の並びかえ状態を表すデータ（遺伝子表現）として使用する。

【００２８】遺伝子表現の例を図８に示す。図において、ＳＭＳコード群８１は、プログラムを構成する複数のストリングのＳＭＳコードの集合である。各ストリングは一つのＳＭＳコードを有し、そのＳＭＳコードは、そのストリング内の複数の命令をそれぞれマージすべき、他ストリング内の複数の位置を示すデータからなる。ＳＭＯコードはプログラムを構成する複数のストリングをマージする順番を指定する。各ストリングのＳＭＳコードは、そのストリングを構成する複数の命令の各々が挿入されるべき、マージ先ストリング内の位置を表すデータからなる。本実施例では同一ストリング内の複数の命令は、他の一つのストリングにマージするように、それらのＳＭＳコードを決めるこれらの位置は、そのマージ先ストリング内のマージ位置の直前に来る命令の番号で表現する。なお、マージ先ストリングの先頭に挿入する場合は命令番号としてコード０を使う。

【００２９】例えば、図８に示した例では、ストリング(Ｂ)のＳＭＳコード(６－６)は、第一命令(７),第二命令(８)共にマージ先ストリング(Ａ)の命令(６)の後に挿入される事を意味する。同様に、ストリング(Ｃ)のＳＭＳ（９－１２）は、第１命令(１０)、第２命令(１２)がそれぞれストリング(Ａ)の命令(９)、(１２)の後に挿入されることを示し、ストリング(Ｄ)のＳＭＳ（１２－１２）は、その第１命令(１４)、第２命令(１６)が共にストリング(Ａ)の命令(１２)の後に挿入されることを意味する。これらのストリング(Ｂ)(Ｃ)(Ｄ)は、いずれもストリング(Ａ)にしか依存関係を有しない。

【００３０】後に説明するように、本実施例では、あるマージすべきストリングが依存関係を有する他のストリングをマージ先ストリングとして使用する。マージすべきストリングが依存関係を有する他のストリングが複数あるとき、それらをマージ先ストリングとして使用する。より具体的には、本実施例では、例えば、図８に示した例では、ストリング(Ｅ)は、図７に関して説明したように、ストリング(Ａ)(Ｂ)に依存する。このストリング(Ｅ)のＳＭＳ（１２）は、ストリング(Ｅ)内の命令（１３）を、実際のマージにあたっては、ストリング(Ａ)に(Ｂ)をマージした後に得られるストリング中の命令（１２）の後に挿入される。図７に関して述べたように、ストリング間の依存関係および各ストリングのレベルはＳＭＤから判別可能である。従って、各ストリングのＳＭＳは、このＳＭＤから決定できる。

【００３１】ＳＭＯコードは、ストリング名の順列からなる。プログラムを構成する複数のストリングをマージして命令列を構成するには、ストリング間の依存関係を満たすようにマージするストリングの順番に制限を設けなければならない。このＳＭＯコードは、このようなストリングをのマージする順を指定する。ストリングのレベルの決め方から分かるように、ある第１のストリングに依存関係のある他の第２のストリングのレベルは、その第１のストリングのレベルより高くなるように決めた。したがって、本実施例では、複数のストリングのマージは、それらのストリングの内、レベルが最低のものに、他のストリングをレベルの低いものから順にマージするように行う。したがって、ＳＭＯ内では、低レベルストリングが高レベルストリングに先行しなければならない。図８のＳＭＯコード８２では、(Ａ)がレベル１、(Ｂ),(Ｃ),(Ｄ)がレベル２、(Ｅ)がレベル３なので、ＳＭＯコード８２は、この制約を守っていることが分かる。

【００３２】ＳＭＯを決定するに必要な情報はＳＭＤから得られることもＳＭＳの場合と同じである。

【００３３】なお、ストリングの切りだし方およびレベル１のストリングの決定には、任意性があることは既に述べたとおりである。本実施例では、あるストリングを他のストリングにマージするときに、当該他のストリング内の先頭の命令より前に、そのマージすべきストリング内の命令もマージすることも可能になっている。勿論、この位置にマージすることがＤＡＧにより許される場合に限る。これにより、マージされるストリング内の命令より前の位置に、マージすべきストリング内の命令をマージ可能になっている。従って、マージするストリングの順序を決めても、マージされる結果は、このマージされる順序に制約されないようになっている。この結果、ストリングのマージ順序を制限しても、その順序に制限されない位置に命令が位置することが可能になっている。

【００３４】以上から分かるように、このように定めた

17

ＳＭＳ群とＳＭＯとの組み合せは、異なる命令の並びに対しては、異なる値を有することが分かる。したがって、特定のＳＭＳ群と特定のＳＭＯとの組み合せが一つの命令の並びを表し、さらに、これらの値を替えることによりいろいろの命令の並び替え状態を表すことが出来る。

【００３５】（３－４）ＳＭＳ群とＳＭＯとの組からの命令列の再生

本実施例では、実行時間の短い命令列（最適あるいはほぼ最適な命令列）に対応するＳＭＳ群とＳＭＯを求めた後、これらから、その命令列を再生する方法を取る。この再生処理は、この最適な命令列の再生以外にも、本実施例ではこの最適な命令列を遺伝アルゴリズムに従って探索する過程でも使用される。すなわち、この探索の過程で候補に上がったＳＭＳ群とＳＭＯとの特定の組を評価するために、その組に対応する命令列を再生し、それらを実際に実計算機で実行させ、その実行時間を測定し、この候補の良し悪しを評価するようになっている。

【００３６】この再生処理手順の理解は、ＳＭＳ群とＳＭＯとの組について上に述べた意義を理解するのに役立つので、ここで再生処理の手順をここで説明する。この再生処理は、図１の適応度評価部１５内のプログラム再生部１５０によりおこなわれる。まず、ＳＭＯで指定される順番にストリングを選択する。すなわち、最初にレベルの最も小さいストリング、実施例ではストリング(Ａ)を選択する。その後に次のレベルのストリングを選択し、後で選択されたストリングをＳＭＳの指示に従って既に選択されたストリングにマージしてゆけば良い。

【００３７】図８に例として示す、ＳＭＯ８２とＳＭＳ群８１とから命令列を再生するには、次のようにすればよい。まずＳＭＯから先頭のストリング(Ａ)を構成する命令列を取り出す。つぎに、ＳＭＤの次のストリング(Ｄ)を取り出し、それを構成する命令（１４）と（１６）との組を、このストリングのＳＭＳ（１２－１２）に従って、ともにストリング(Ａ)の命令（１２）の後にマージする。実際には、まず、命令（１４）を命令（１２）の後にマージし、その後に命令（１６）をマージする。すなわち、同じストリングに属する命令（１４）と（１６）の順序は入れ換えない。こうして新たな第１のストリング(１)—(２)—(３)—(４)—(５)—(６)—(９)—(１１)—(１２)—(14)—(16)—(１８)—(１７)を得る。

【００３８】ついで、ＳＭＯの次のストリング(Ｃ)を取り出し、そのストリングを構成する命令（１０）と（１５）を、ＳＭＳ（９－１２）に従って、この得られた第１の新たなストリングのそれぞれ命令（９）、（１２）の後にマージする。こうして第２の新たなストリング(１)—(２)— ... —(９)—(10)—(１１)—(１２)— ＊(15)—(１４)—(１６)—(１８)—(１７)をえる。同様にしてＳＭＯの次のストリング(Ｂ)を取り出し、それを構成する命令（７）、（８）を、ＳＭＳ（６－６）に従って、この第２の新たなストリングの命令（６）の後にマージして(１)—(２)— ... —(６)—(7)—(8)—(９)—(１０)— ... —(１６)—(１８)—(１７)をえる。最後にストリング(Ｅ)を構成する命令（１３）を、このＳＭＳ（１２）に従って、第３の新たなストリングの命令（１２）の後にマージして最終的なストリング(１)－(２)－(３)－(４)－(５)－(６)－(７)－(８)－(９)－(１０)－(１１)－(１２)－(13)－(１５)－(１４)－(１６)－(１８)－(１７)を得る。こうしてＳＭＳ群とＳＭＯとから、命令列の並びが、命令番号の列として決定された。この決定された並びにしたがって、プログラム再生部１５０は、それぞれの命令番号を有する命令をもとのアセンブリプログラム２（図１）より取り出すことにより実際に並び替えた命令列からなるプログラムを再生する。

18

【００３９】（３－５）ＳＭＳ群とＳＭＯの取り得る値

ＳＭＯで規定されるストリングのマージ順序のうち、同一のレベルのストリングのマージ順序は変更しても、レベルの低いストリングにレベルの高いストリングをマージするべきと言う上記制約には反しない。従って、同じレベルのストリング、この実施例では、ストリング(Ｂ)(Ｃ)(Ｄ)、の順序を入れ換えて複数のＳＭＯを作ることが可能である。具体的には、本実施例では、以下のような６つのＳＭＯを使用することが出来る。

【００４０】各ストリングの命令の他のストリングへのマージ位置は、ＤＡＧで規定される命令間の依存関係により定まる許容範囲内で変えることができ、それに応じて異なるマージ状態すなわち命令並び替え状態を実現できる。従って、各ストリングのＳＭＳコードも、これらの異なる並び替えに対応していろいろの値を持ち得る。例えば、ストリング(Ｂ)の命令(７)(８)については、ＤＡＧ上命令(７)が(８)より先に実行されねばならず、命令(８)が(９)より先に実行されねばならないという制約がある。従って、ストリング(Ｂ)の命令(７),(８)は、ストリング(Ａ)上の先頭から命令(６)の後までの範囲にマージ可能である。従って、ストリング(Ｂ)の、ストリング(Ａ)に対するＳＭＳコードは、０－０から６－６の範囲（ただし後者の番号は前者の番号以上）で変更可能である。

【００４１】同様に、ストリング(Ｅ)の命令(１３)のストリングに対するＳＭＳコードは、(Ｂ)を(Ａ)にマージした結果のストリング上の命令(７)から(１８)の範囲で変えられる。

【００４２】

【表１】

表１

ＳＭＯ＃１　(Ａ)(Ｂ)(Ｃ)(Ｄ)(Ｅ)
ＳＭＯ＃２　(Ａ)(Ｂ)(Ｄ)(Ｃ)(Ｅ)
ＳＭＯ＃３　(Ａ)(Ｃ)(Ｂ)(Ｄ)(Ｅ)
ＳＭＯ＃４　(Ａ)(Ｃ)(Ｄ)(Ｂ)(Ｅ)
ＳＭＯ＃５　(Ａ)(Ｄ)(Ｂ)(Ｃ)(Ｅ)
ＳＭＯ＃６　(Ａ)(Ｄ)(Ｃ)(Ｂ)(Ｅ)

ここで、ＳＭＯの番号はこれらの異なる値を識別するために便宜的に表示するものであって、ＳＭＯ自体はあくまでストリングの名称の列からなる。

【００４３】（３－６）最良遺伝子表現探索部１４

図１に戻り、最良遺伝子表現探索部１４の処理を以下に述べる。ここでは遺伝的アルゴリズムとして、ブリーダーアルゴリズム(Breeder Algorithm)(参考文献[3])と呼ぶアルゴリズムを用いている。この遺伝的アルゴリズムは、生命体の進化・適応過程の数学的モデルであり、適当な遺伝子で表現される個体集団の各世代に対して、交配(交叉)、突然変異、局所探索(後天的適応)などの遺伝的操作を施すことにより、世代を追って集団としての改良を図って行くものである。本実施例は、このアルゴリズムを命令のスケジュールに使用する。従って、以下でも個体、世代などのような遺伝的アルゴリズムで使用される用語を用いる。以下では、個体は命令列の特定の並びを表し、その個体を特定するデータ（遺伝子表現）は、既に述べたＳＭＳ群とＳＭＯとの組で表す。

【００４４】（３－６－１）初期世代の生成処理１４１

まず始めにＤＡＧ１２１とＳＭＤ１３１を用いて初期世代のＮ個の個体（今の例では命令列の並び）の遺伝子表現１４０１（これはＳＭＳ群とＳＭＯとからなる）の生成を行う。併せて、それぞれの個体の適応度１４４１を計測する。各個体の各ストリングのＳＭＳコードは、ＳＭＤ（図７）で決まるマージ先とＤＡＧ１２１で規定される命令のマージ位置の許容範囲に基づいて乱数を用いて決める。具体的には各個体の各ストリングのＳＭＳコードの発生にあたっては、プログラムを構成するストリングをそれらのレベル順に上記したＳＭＤ（図７）から取り出す。先頭のレベル１のストリング、今の例では(Ａ)、を取り出したときには、このストリングはＳＭＳを要しないので、以下の処理をしないで、つぎのストリングを上記ＳＭＤから取り出す。取り出されたストリングを構成する複数の命令をそのストリング内の並び順にＳＭＤから取り出す。取り出された命令の、そのストリングのマージ先ストリング（これは図７に示されるように、ＳＭＤにより定まる）内の位置を、ＤＡＧ１２１で許容される範囲内から乱数により決定する。この処理をストリング内の命令が尽きるまで繰り返す。

【００４５】たとえば図７のＳＭＤで示されるストリングの場合、レベル１のストリング(Ａ)はＳＭＳコードが不要なのでスキップし、まずレベル２の(Ｂ)に取り組む。このストリングのマージ先ストリングは、図７に示すＳＭＤからストリング(Ａ)である。ストリング(Ｂ)の第１命令(７)のマージ位置の許容範囲は、図５のＤＡＧ１２１から、マージ先ストリング(Ａ)上の命令(１)から(６)の間なので、その範囲から一つのマージ位置の命令、たとえば命令(２)をランダムに選択する。第２命令(８)のマージ位置の許容範囲はストリング(Ａ)上の、命令(７)用に選択した命令(２)から(６)までの間であることがＤＡＧ１２１から判別される。従って、この命令(８)のマージ位置として、その許容範囲から一つのマージ位置、たとえば命令(５)をランダムに選択する。この結果、この個体に関する命令(Ｂ)のＳＭＳコードが(２－５)と決定された。続いてストリング(Ｃ),(Ｄ)のＳＭＳコードを同様に決定する。最後に(Ｅ)の命令(１３)のＳＭＳコードを(Ａ)に(Ｂ)をマージしたストリング(１)—(２)—(7)—(３)—(４)—(５)—(８)—(６)— ... —(１８)—(１７)上の命令(７)から(１８)の間でランダムに選択する。

【００４６】各個体のＳＭＯコードは、例えば、図８に示した例では、スケジュールすべきプログラムを構成するストリング(Ａ)(Ｂ)(Ｃ)(Ｄ)(Ｅ)の内、同一レベルに属するストリング（今の例では(Ｂ)(Ｃ)(Ｄ)）の範囲内でこれらのストリング（今の例では(Ｂ)(Ｃ)(Ｄ)）の並びを乱数を用いて決める。すなわち、先に説明したＳＭＯがとり得る６つのＳＭＯのいずれかを乱数で選ぶことになる。以上の結果、一つの個体のＳＭＳ群とＳＭＯが決定された。このような試行をあらかじめ規定された個体数であるＮ回分繰り返してＮ個の個体に対する遺伝子表現１４０１を生成する。

【００４７】その後、各個体について適応度を計測するために、適応度評価部１５を起動する。適応度として、各遺伝子表現が表す命令列の並びを有するプログラムの実行時間を使用する。勿論実行時間が少ない遺伝子表現の方が適応度が高い。この実行時間のため、適応度評価部１５内のプログラム再生部１５０が、各個体の遺伝子表現から、それが表す命令列を有するプログラムを再生する。その手順は既に述べたとおりである。さらに、実行部１５１が、この再生された命令列は実計算機２０００（図２）により実際に実行させ、実行完了に要した時間を計測する。各個体に関する実行時間の計測結果は、適応度データ１４４１として、後の使用のために記憶する。

【００４８】続いて世代を追って個体を改良してゆく。

【００４９】（３－６－２）局所探索処理１４２

始めに上で得られたＮ個の個体に対して得られたＮ個の遺伝子表現１４０１に対して、局所探索による改良を行

い、改良後の新たな遺伝子表現１４０２を生成する。局所探索１４２では、Ｎ個の初期世代の個体のそれぞれの遺伝子表現を構成するＳＭＳコード群ないしＳＭＯコードのいずれか一つのコード値を隣接値に変更し、変更前の個体の適応度より高い適応度を有する個体が見つかれば、その変更後の個体でもって、変更前の個体を置き換える。上記隣接値への変更を、その個体のいろいろのストリング、ＳＭＯ内のいろいろのレベルの部分に対して繰り返して行く。ここで、あるストリングのＳＭＳコードの隣接値とは、そのストリング内の特定の命令をマージする、マージ先ストリング上の、そのＳＭＳが指定している位置を、隣接する位置に変更したときのＳＭＳである。この隣接位置は、そのマージ先ストリング上の上流側の隣接命令（もとのＳＭＳが指定するマージ位置にある命令より先に実行すべき命令）もしくは下流側の隣接命令のいずれかの位置である。但し、この隣接値への変更は、その隣接値にその特定の命令をマージすることが命令依存関係により許されている場合にのみ行なう。図９(ａ)は、局所探索操作により、ＳＭＳコードを隣接値に変更した例で、ここではストリング(Ｃ)の第一命令である(１０)のＳＭＳコード９を、ストリング(Ｃ)のマージ先ストリング(Ａ)内の、そのＳＭＳが指定する命令(９)の上流側で隣接する命令(６)に対する隣接値６に変更している。

【００５０】ＳＭＯの隣接値とは、ＳＭＯが示すストリングの並びを、辞書の見出し順に並べたときの隣接するＳＭＯのことである。隣接位置は、見出しの上流側で隣接する位置（見出しに先に表れる位置）あるいは下流側で隣接する位置のいずれかである。本実施例で使用している命令列の例では、ＳＭＯは６つの値を取ることが出来、それらは先に、表１にＳＭＯ＃１からＳＭＯ＃６として略記して示した。そこで示した６つのＳＭＯの値の並びは辞書の見出し順になっている。図９(ｂ)は、ＳＭＯを隣接値に変更した例を示す。ここで、レベル２のストリング(Ｃ)と(Ｂ)の順番を入れ替え、左側のＳＭＯ（これは表１のＳＭＯ＃１に相当する）をそれに隣接するＳＭＯ＃２に変更している。

【００５１】以下具体的に、この局所探索の処理を説明する。なお、局所探索で変更すべきＳＭＳあるいはＳＭＯ内の部分を決定する手順は、実質的に初期個体の生成の場合の同じである。まず各個体について、そのストリングをレベルの順に一つづつ選択し、選択したストリングの各々についてＳＭＳの変更に関する以下の処理を行なう。但し、レベル１のストリングはＳＭＳを有しないので、以下の処理は不用であるからである。次に、選択された一つのストリングを構成する複数の命令を順に一つづつ選択する。これらの命令を選ぶ順は、それらの命令のそのストリング内での順に従う。

【００５２】こうして現在選択されている一つのストリングと、そのストリングから現在選択されている一つの命令に関するＳＭＳコードを、現在選択されている命令の現在のマージ位置の上流側の隣接するマージ位置に対するＳＭＳへの変更を試みる。すなわち、その隣接位置への変更がＤＡＧにより許されるかを判断し、許されるならば、この変更を行ない、許されないときにはこの変更は行なわない。こうして、選択されたストリングの変更後のＳＭＳと他のストリングの未変更のＳＭＳとからなるＳＭＳ群と、もとのＳＭＯとの組からなる新たな個体を得る。この新たな遺伝子表現の適応度の評価を、先に初期世代の個体の場合と同様にして行なう。上のＳＭＳの変更と同じことを、上記現在選択されている命令の現在のＳＭＳが指定するマージ位置の下流側で隣接するマージ位置に対するＳＭＳへの変更を同じように行なう。

【００５３】さらに、以上を、この選択されたストリングの他の命令に関して順次繰り返す。この繰返しのときには、先に実行されたＳＭＳの変更は、無視し、新たな変更は、操作を受けている個体の最初のＳＭＳ群に対して行なう。このことはこの局所操作に関する以下の処理でも同じである。さらに、以上と同じことをストリングを変えて行なう。こうして、レベル２以上の全てのストリングの全ての命令に対して以上のことが行なうことによりＳＭＳの変更は終了する。

【００５４】例えば、図８のストリング群の例では、レベル１のストリング(Ａ)はＳＭＳをもたないのでスキップし、まずレベル２のストリングの一つ（Ｂ）の最初の命令７に対するＳＭＳコードである６を、現在のマージ先命令６の上流側の隣接の命令５をマージ位置とするＳＭＳへ変更する。この変更は、ＤＡＧにより許され得る変更である。続いて、同じストリングの同じＳＭＳを、命令６の下流側の命令９をマージ位置とするＳＭＳに変更することを試みるが、命令７は、ＤＡＧにより、命令９より先に実行されねばならないから、このマージ位置の変更は行なわない。ついで、ストリング(Ｂ)の次の命令８に関するコード６の変更を試みる。ついでストリング(Ｃ)，(Ｄ)，(Ｅ)の順に進む。なお、ストリング(Ｃ)の最初のＳＭＳコード９を６に変更する処理の結果は、図９(ａ)に示したとおりである。

【００５５】この後は、ＳＭＯを隣接値に変更する。ＳＭＯの隣接値への変更は、ＳＭＯをレベルの低いグループの順に行なう。例えば、レベル１のストリングが複数ある場合には、他のレベルのストリングののマージ順は変更しないで、レベル１のそれらの複数のストリングのマージ順を、辞書見出しの上流側に隣接するＳＭＯに対するものに変更する。現在処理中の個体の元のＳＭＳ群と、この変更後のＳＭＯとの組により表される新たな個体を得る。この新たな個体について適応度を、前述の方法により評価する。以上を辞書の見出しの下流側で隣接するＳＭＯに対するものに変更する。その後、レベル２以降のレベルの各々について同様の処理をする。もし、

レベル１のストリングが一つしかない場合には、レベル１に関して上に説明した処理を行なうことなく、レベル２以降のストリングの処理に移る。

【００５６】例えば、図８のストリングの例では、ＳＭＯの第１レベルグループのストリング（Ａ），第２レベルグループのストリング列(Ｄ)－(Ｃ)－(Ｂ)，第３レベルグループの(Ｅ)の順にそれぞれのグループ内のストリングの入れ替えを試みるが、この例では、ストリングの入れ換えが可能な複数のストリングを有するグループは第２レベルグループのみである。この第２グループのストリング列(Ｄ)－(Ｃ)－(Ｂ)を上流側で隣接するストリング列(Ｄ)－(Ｂ)－(Ｃ)をで置き換えることにより、隣接するＳＭＯを得る。なお、元の第２レベルグループのストリング列(Ｄ)－(Ｃ)－(Ｂ)に対する辞書の見出し下流側で隣接するストリング列はない。こうして、一つの個体の局所変形個体が全て得られた後、それらの局所変形個体と元の処理中の個体の適応度とを比較し、最も適応度の改良の大きい個体を元の個体の改良個体として選択し、元の個体の代りに使用する。

【００５７】その後、他の個体について同様のことをする。こうして適応度の改善が図られたＮ個の個体１４０２とそれらの適応度１４４２を得る。

（３－６－３）交配プールの生成１４３

このようにして得られたＮ個の個体から、適応度が上位にある一定数Ｍ個の個体を選択して交配プールを構成する。この数Ｍは適宜決定する。例えば、Ｎ／２でもよい。この交配プールからからランダムにＮ／２対の個体１４０３を選ぶ。この際、交配プール中の各個体はランダムに選択されるため、それぞれが選択される回数は各個体により異なる。併せてそれらの適応度１４４３を記憶する。交配プールの構成法には、各個体に適応度に応じた選択確率を与え、それを用いて局所探索の結果１４０２から確率的に選択するという方法もある(参考文献[２])。

【００５８】（３－６－４）交叉操作１４４

ここでは各対に交叉操作を施し、新たなＮ個の個体１４０４を生成し、それらの適応度１４４４も記憶する。交叉操作では、各対の二つの個体の間で、特定ストリングのＳＭＳコードないしＳＭＯコードの内の特定レベルの部分を交換し、新たな個体対を生成する。そのために、各個体対について、以下の処理を行なう。

【００５９】まずＳＭＳコードを交換するか、ＳＭＯコードを交換するかの交換対象の選択を乱数により行なう。ＳＭＳコードの交換が選択されたとき、交換を行うべきストリングを選択する。その選択に当たっては、

（条件１）個体対の間でそのストリングのＳＭＳコードが異なること、すなわち、交叉により親とは別の対が生成されること

（条件２）健全な個体の生成を保証するためにそのストリングのマージ先のストリング系列のＳＭＳコードが全て、その個体対の間で同じである

という二つの条件を満たすストリングを選択する。このような条件を満たすストリングが複数あるときには、それらのストリングの中から乱数により一つのストリングを選択する。選択されたストリングのＳＭＳを処理中の個体対の間で交換する。さらに、選択されたストリングをマージ先とするストリング群があるときには、それらのＳＭＳコードも同時に交換する。

【００６０】例えば、図１０の(ａ)の上部に示す個体対の間では、ストリング(Ｂ)，(Ｃ)共にこれらの個体対の間でＳＭＳコードが異なる。すなわち、ストリング(Ｂ)のＳＭＳは６－６と４－７、ストリング(Ｃ)のＳＭＳは９－１２と５－９である。さらに、ストリング(Ｂ)，(Ｃ)共、マージ先ストリングは(Ａ)であり、そのＳＭＳコード(なし)が共通である。従って、これらのストリング(Ｂ)，(Ｃ)はともに上記二つの条件を満たすため、交換の候補となる。ストリング(Ｃ)が交換対象に選択され、ＳＭＳコードを交換した場合、図(ａ)の下部に示す新たな個体対が得られる。一方、仮にストリングが(Ｂ)が交換対象に選択されたときは、ストリング(Ｅ) がこのストリング(Ｂ)をマージ先ストリングとするため、このストリング(Ｅ) のＳＭＳコードも併せて個体対の間で交換する。

【００６１】上記の交換対象の選択において、ＳＭＯコードの交換が選択された時は、個体対の間でＳＭＯコードの内の交換すべき特定レベルを選択する。その選択においては、個体対の間でＳＭＯコードの内、そのレベルの部分が異なることを条件とする。この条件を満たすレベルが複数存在するときは乱数を用いて一つのレベルを選択する。ＳＭＯコードのうち、こうして選択されたレベルの部分を交換すればよい。例えば、図１０(ｂ)では、図の左側に示すように、個体対の間で、元のＳＭＯコードのレベル２部分Ｂ－Ｃ－ＤとＢ－Ｄ－Ｃが相互に異なり、交換可能であり、交換の結果、図の右側に示すＳＭＯコードが得られる。

【００６２】以上において、処理中の個体対についてＳＭＳを交換すべきストリングまたはＳＭＯ内部分データが見つからないときには、元の個体対をそのまま交叉結果として使用する。このようにして、各個体対にたいして新に個体対が生成される。以上を、他の個体についても行なう。こうして、新にＮ個の個体１４０４が決定される。交換操作前のＮ個の個体の代りに使用される。

【００６３】（３－６－５）突然変異操作１４５

続いて交叉操作の結果得られたＮ個の個体の各々に対して突然変異操作を施し、Ｎ個の新たな遺伝子表現集合１４０５とそれらの適応度１４４５を生成する。突然変異操作は各個体のＳＭＳコード群の各々およびＳＭＯコードの各レベルに対応する部分の値をＤＡＧから決まる制約条件の範囲内で確率的に変化させる。変化させる対象を選択する手順は初期世代の生成あるいは局所探索のと

きの変更箇所の選択手順と同様である。概略的には、、まずＳＭＳコードをストリングのレベルの順、ストリングの中ではそれを構成する命令の順に選択する。ついでＳＭＯコードをレベルグループの順に選択する。

【００６４】この突然変異操作では、以下の点で局所操作と異なる。変更させる対象を選択する毎に、その対象を突然変異させるか否かを、乱数により確率的に決定する。さらに、この対象を変化させると決定された場合、その対象を変化させる範囲は、局所操作における前述の隣接値に限らない。例えば、変更させる対象がＳＭＳコードである場合で、かつ、それを突然変異をさせると決定した場合、変更後のＳＭＳを、ＤＡＧの制限範囲内でランダムに決める。

【００６５】さらに、ＳＭＯコードのあるレベルの部分を変更する場合も、その部分が取り得る値の範囲の中でランダムに決める。さらに、ある変更対象に対して変更を加えた後、他の対象に変更を加えるときには、先の対象に変更を加えて得られる新たな個体内の当該新たな対象に対して、変更を加える。なお、ＳＭＳコードを変更する場合、変更後のＳＭＳコードを、ＤＡＧにより許容される範囲内で決めることは、局所探索の場合と同じである。

【００６６】たとえば、図１１の(ａ),(ｂ)に突然変異操作の例を示す。(ａ)はある個体のストリング(Ｃ)の第１の命令(１０)のＳＭＳコードの値９を突然変異させた例で、ここでは、このＳＭＳを、マージ先ストリング(Ａ)の中の、ＤＡＧ(図５)で規定される、この命令(１０)をマージ可能な範囲((１)―(２)―(３)―(４)―(５)―(６)―(９)の範囲)に属する命令５に対するものに変化させている。さらに、同図(ｂ)はＳＭＯコードのレベル２の部分を突然変異させる例で、この部分をＢ－Ｃ－ＤからＤ－Ｂ－Ｃに変化させている。

【００６７】こうして得られた新たな個体の適応度を評価する。

【００６８】以上を他の個体にも行なう。こうして、新にＮ個の個体１４０５とそれらの適応度１４４５を得る。

【００６９】(３－６－６)世代交替１４６

最後に世代交代１４６を行い、新世代の遺伝子表現集合１４０６を生成する。すなわち、基本的には旧世代の個体を全て新世代で置き換えるが両世代の最良個体は旧世代に属していても残す。そのために突然変異操作１４５で得られたＮ個の個体１４０５の適応度１４４５と旧世代の最良のＮ個の個体１４０２の適応度１４４２と比較し、適応度が高いＮ個の個体１４０６を第２世代の個体として選択する。それらの適応度１４４６も併せて記憶する。その後、この第２世代の個体１４４６とその適応度１４４６を初期世代の個体１４０１とその適応度１４４１の代わりに使用して、この第２世代のために局所探索１４２以降の処理からなる世代処理を繰り返す。この繰返し前に、各世代の個体適応度の最大値を記憶する。今回生成された新たな世代の個体の適応度の最大値と、このように記憶された過去の世代の適応度の最大値から、世代処理による最良個体の改善が進まなくなったか否かあるいは進みの程度を判別し、進まなくなったときあるいは進みが極度に遅くなったときに、世代処理を終了ないし完了する。世代処理の繰返しを打ち切った後、最新の世代中で最も適応度の高い遺伝子表現を選択し、再度プログラム再生部１６により最良アセンブリプログラム３を再生する。

【００７０】(変形例)以上の実施例は以下のように変形することも可能である。

(１)局所探索１４２の処理において、各個体の局所変形をその個体に関して可能な全ての局所変形個体を生成する代わりに、その個体の局所変形の過程でその個体より適応度が高い個体が得られた時点で、その個体の局所変形を打ち切る方法もある。この方法によれば各個体の局所変形個体の生成数が減少し、それだけ処理時間が低減できる。勿論、実施例のように、各個体について、生成可能な全ての局所変形個体を生成する方が、より適応度の高い個体を生成する可能性が高まる。

【００７１】(２)局所探索処理自体をスキップすることも可能である。その場合、上記変形例(１)よりさらに処理時間が短縮される。

(３)局所探索操作、交叉操作、突然変異操作を施す順序は必ずしも上記に限らず、他の順序であって良い。

(４)最良遺伝子表現探索部１４が使用する遺伝的アルゴリズムとしては、実施例で使用した参考文献[３]記載のアルゴリズムに代えて、参考文献[２]記載のアルゴリズムを用いても良い。

(５)交配プールの構成法は実施例記載の方法に代えて、局所探索処理１４２で得られたＮ個の個体から、それらの適応度を用いて確率的に選択する方法が考えられる(参考文献[２])。

【００７２】(６)上記の実施例において、スケジュールを施す命令列はアセンブリプログラムの替りに機械語プログラムであっても良い。この場合は図１の２、３はそれぞれ最良機械語プログラム、最良アセンブリプログラムとなる。

(７)各個体から再生されたアセンブリプログラムを評価するにあたり、実施例のごとく、元のプログラムを実行予定の実計算機で実際にその再生されたプログラムを実行する代わりに、その計算機の構造と動作を模擬して、この実行時間を推定するシミュレータプログラムを使用してもよい。

(８)命令スケジューラ１１がコンパイラ１０の内部に取り込まれていてもよい。

【００７３】(実験例)図１２には仮想的なＲＩＳＣマシンと図４のアセンブリプログラムを対象としておこなった１０世代分の試行による適応度の改善状況を個体数

１０と２０、局所探索あり／なしのケースについて示す。この命令列の最良適応度は６９マシンサイクルであることが知られており、それは個体数(NPOPL)２０の場合に第４世代(局所探索なし)あるいは第１世代(局所探索あり)で発見された。一般に局所探索ありのほうが改善が早く進む。このように最良個体の探索が少ない世代の試行で行える。図１３にそれぞれの場合での、最良適応度を持つ個体への到達履歴を示す。各個体は左から順にストリング(Ｂ),(Ｃ),(Ｄ),(Ｅ)のＳＭＳコードを表示して示す。ＳＭＯコードは省略する。丸括弧内の数字は各個体の適応度を示す。矢印は遺伝的操作による個体間の遷移を示す。ｒは交叉操作、ｍは突然変異操作、ｌは局所探索操作、ｓは最良個体の世代送り操作である。

【００７４】この図から、本実施例の遺伝子表現と遺伝的操作が適応度の改善に有効に作用していることが判る。局所探索なしの場合の最良個体の遺伝子から命令列を再現すると、(１)－(１０)－(２)－(１５)－(１４)－(３)－(７)－(４)－(１３)－(５)－(１６)－(８)－(６)－(９)－(１１)－(１２)－(１７)－(１８)となる。一方、例題の命令列を文献[１]に記載された従来方式により入れ替えると、(１０)－(１)－(７)－(２)－(８)－(３)－(１４)－(４)－(１６)－(５)－(１３)－(６)－(１５)－(９)－(１１)－(１２)－(１７)－(１８)となり、その適応度は７４サイクルで本発明を用いた場合の適応度６９サイクルに及ばない。その理由は図５に示すＤＡＧ上そこから派生する線(アーク)数の多い命令(１０)を命令(１)に先行してスケジュールすることにあり、固定的規則の限界を示している。また他の例題では、並べ替え前の命令列の実行サイクル数１６２１に対し従来方式の並べ替えでは１６１６サイクル、本発明の並べ替えでは１１７１サイクルとの結果が得られている。

【００７５】

【発明の効果】本発明では、命令列のスケジュールに適した遺伝子表現と遺伝的操作の導入により従来方式より優れた命令列への並べ替えが実現できる。また、本発明では、計算機の設計情報を必要としないで命令列をスケジュールすることが出来るので、計算機特性の変化に自動的に追従して命令列をスケジュールできる。

【図面の簡単な説明】

【図１】本発明の実施例による命令列スケジュール処理の流れ図。

【図２】図１の処理を実行するための計算機システムの全体構成図。

【図３】図１のストリングマージ図（ＳＭＤ）生成部（１３）の処理の流れ図。

【図４】図１の図１のアセンブリプログラム（２）の例を示す図。

【図５】図４のアセンブリプログラムに対して図１のＤＡＧ生成部（１２）により得られる命令依存解析図（ＤＡＧ）を示す図。

【図６】図４のアセンブリプログラムに対して図１のＳＭＤ生成部（１３）により得られたストリングベクトル（ＳＴＶ）を示す図。

【図７】図４のアセンブリプログラムに対して図１のＳＭＤ生成部（１３）により得られたストリングマージ図（ＳＭＤ）の例を示す図。

【図８】図４のアセンブリプログラムに対して図１の初期世代生成処理（１４１）により得られた遺伝子表現（ＳＭＳコードとＳＭＯコードの組）の例を示す図。

【図９】図１の局所探索操作（１４２）により得られた個体の例を示す図。

【図１０】図１の交叉操作（１４４）により得られた個体の例を示す図。

【図１１】図１の突然変異操作により得られた個体の例を示す図。

【図１２】図１の最良遺伝子探索処理（１４）よる１０世代にわたる探索の履歴を示す図。

【図１３】図１の最良遺伝子探索処理（１４）よる最良命令列の１つへの到達の履歴を示す図。

【符号の説明】

１２１…命令解析図（ＤＡＧ）、１３１…ストリングマージ図（ＳＭＤ）、３１１…ストリングベクタ（ＳＴＶ）、８１…ストリングマージ状態（ＳＭＳ）コード、８２…ストリングマージ順序（ＳＭＯ）コード。

【図６】

図 6

311 ストリングベクトル（ＳＴＶ）

(A)：1－2－3－4－5－6－9－11－12－18－17

(B)：7－8

(C)：10－15

(D)：14－16

(E)：13

【図７】

図 7

131 ストリングマージ図（ＳＭＤ）

(A)：レベル1：1－2－3－4－5－6－9－11－12－18－17

(B)：レベル2：7－8

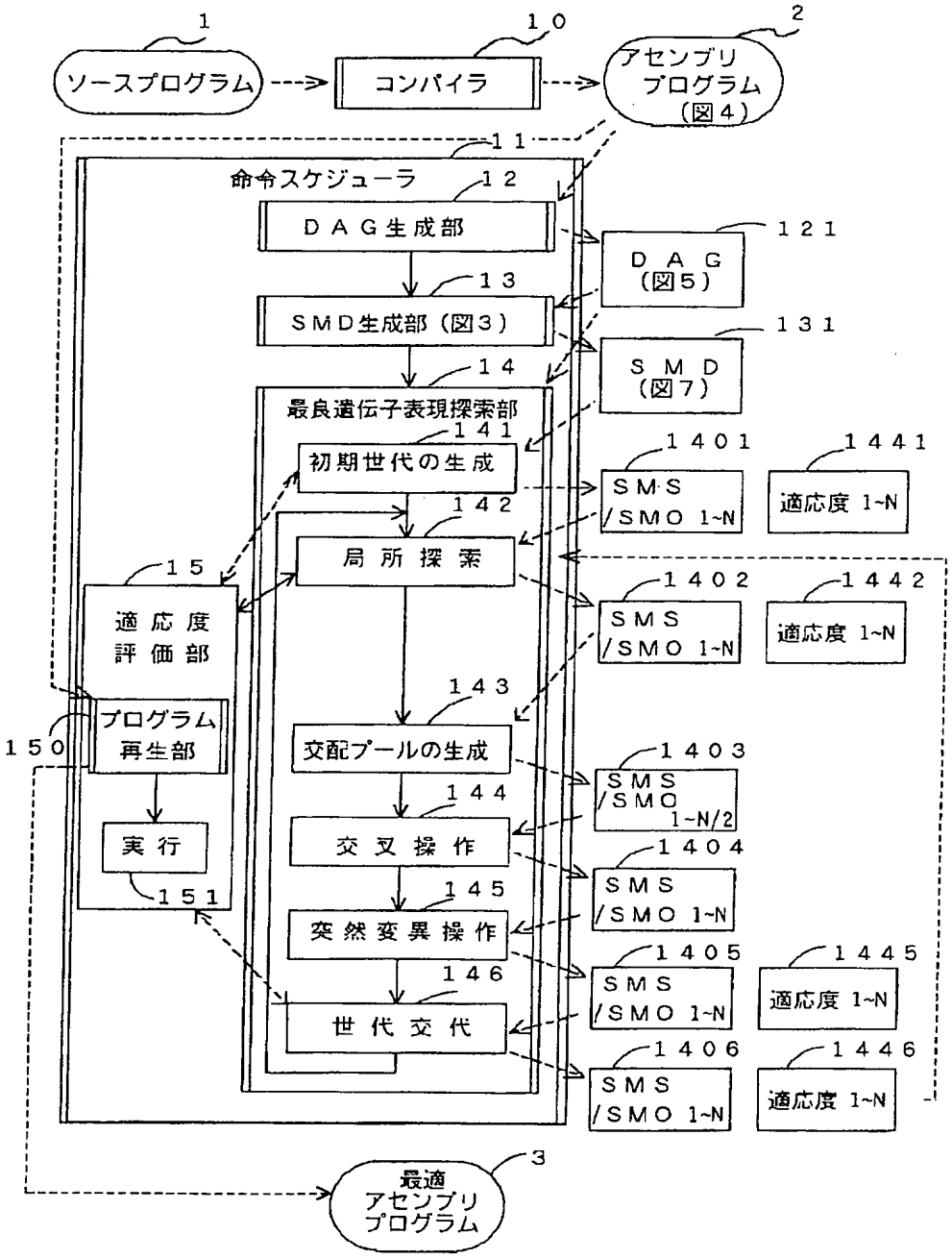
(C)：レベル2：10－15

(D)：レベル2：14－16

(E)：レベル3：13

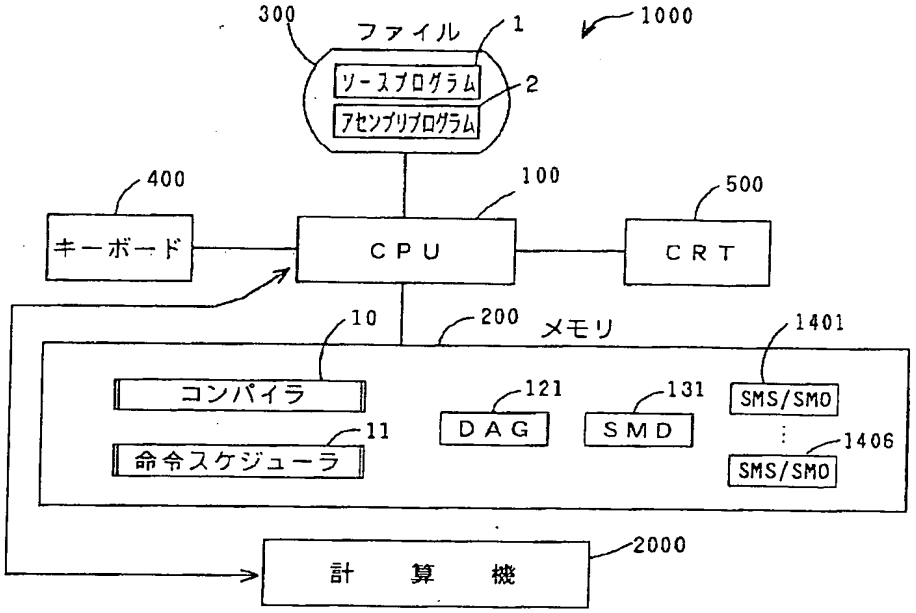
【図１】

図１


1
ソースプログラム
10
コンパイラ
2
アセンブリ
プログラム
（図４）
11
命令スケジューラ
12
ＤＡＧ生成部
121
ＤＡＧ
（図５）
13
ＳＭＤ生成部（図３）
131
ＳＭＤ
（図７）
14
最良遺伝子表現探索部
141
初期世代の生成
1401
ＳＭＳ
／ＳＭＯ 1~N
1441
適応度 1~N
142
局所探索
1402
ＳＭＳ
／ＳＭＯ 1~N
1442
適応度 1~N
15
適応度
評価部
143
交配プールの生成
1403
ＳＭＳ
／ＳＭＯ
1~N/2
150
プログラム
再生部
144
交叉操作
1404
ＳＭＳ
／ＳＭＯ 1~N
実行
151
145
突然変異操作
1405
ＳＭＳ
／ＳＭＯ 1~N
1445
適応度 1~N
146
世代交代
1406
ＳＭＳ
／ＳＭＯ 1~N
1446
適応度 1~N
3
最適
アセンブリ
プログラム

【図２】

図　２


300
ファイル
1
1000
ソースプログラム
2
アセンブリプログラム
400
100
500
キーボード
CPU
CRT
10
200
メモリ
1401
コンパイラ
121
131
SMS/SMO
DAG
SMD
11
1406
命令スケジューラ
SMS/SMO
2000
計　算　機

【図５】

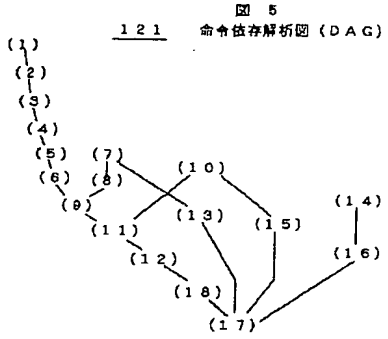
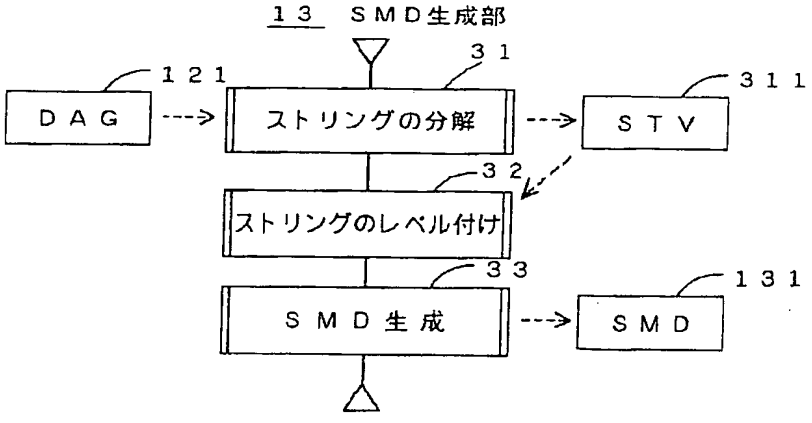


図　５
１２１　命令依存解析図（ＤＡＧ）
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13)
(14)
(15)
(16)
(17)
(18)

【図３】

図　３


１３　ＳＭＤ生成部
３１
１２１
３１１
ＤＡＧ
ストリングの分解
ＳＴＶ
３２
ストリングのレベル付け
３３
１３１
ＳＭＤ生成
ＳＭＤ

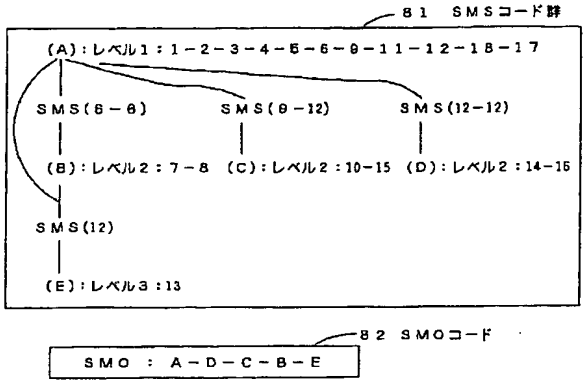
【図４】

図　４

```
 1   L15:sll    %g3,    4, %o0
 2       sub    %o0, %g3, %o0
 3       add    %o0, %o5, %o0
 4       add    %o0, %g1, %o0
 5       add    %o0, %l2, %o0
 6       ldd    [%o0+%o4], %f2
 7       add    %o3, %l1, %o6
 8       ldd    [%o6+%o5], %f0
 9       fmuld  %f2, %f0, %f2
10       add    %o1, %l3, %o7
11       ldd    [%o7+%o4], %f0
12       faddd  %f2, %f0, %f2
13       add    %o3, 2400, %o3
14       add    %o2, 2400, %o2
15       add    %o1, 2416, %o1
16       cmp    %o2, %l0
17       ble    L15
18       std    %f2, [%o7+%o4]
```

【図８】

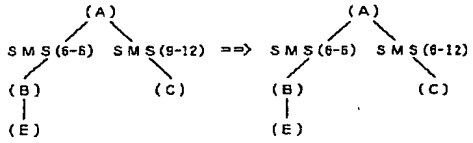
図８



【図９】

図　９

(a) 局所探索作操作１：ストリング（c）のＳＭＳコードの隣接値への更新



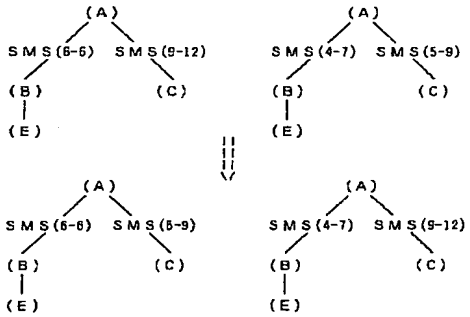
(b) 局所探索操作２：レベル２のＳＭＯ部分コードの隣接値への更新

A－D－C－B－E　==>　A－D－B－C－E

【図１０】

図１０

(a) 交叉操作１：ストリング（c）のＳＭＳコードの交換



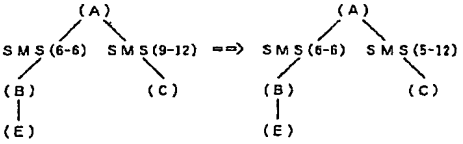
(b) 交叉操作２：レベル２のＳＭＯ部分コードの交換

A－B－C－D－E　==>　A－B－D－C－E
A－B－D－C－E　==>　A－B－C－D－E

【図１１】

図１１

(a) 突然変異操作１：ストリング（c）のＳＭＳコードの更新



(b) 突然変異操作２：レベル２のＳＭＯ部分コードの更新

A－B－C－D－E　==>　A－D－B－C－E

【図１２】

図 １２

（ａ）局所探索なし

NPOPL = 10

gs 0	min	73	max	129	avr	93
1	min	73	max	89	avr	82
2	min	71	max	95	avr	80
3	min	71	max	91	avr	77
4	min	71	max	73	avr	72
5	min	71	max	73	avr	72
6	min	71	max	80	avr	72
7	min	71	max	71	avr	71
8	min	71	max	80	avr	71
9	min	71	max	71	avr	71
10	min	71	max	71	avr	71

NPOPL = 20

gs 0	min	71	max	129	avr	91
1	min	71	max	105	avr	83
2	min	71	max	90	avr	78
3	min	71	max	80	avr	74
4	min	69	max	86	avr	73
5	min	69	max	87	avr	72
6	min	69	max	79	avr	71
7	min	69	max	74	avr	70
8	min	69	max	81	avr	71
9	min	69	max	74	avr	70
10	min	69	max	73	avr	69

（ｂ）局所探索あり

NPOPL = 10

gs 0	min	73	max	118	avr	87
1	min	71	max	80	avr	74
2	min	71	max	74	avr	73
3	min	71	max	75	avr	72
4	min	71	max	75	avr	72
5	min	71	max	73	avr	72
6	min	71	max	73	avr	71
7	min	71	max	71	avr	71
8	min	71	max	71	avr	71
9	min	71	max	71	avr	71
10	min	71	max	71	avr	71

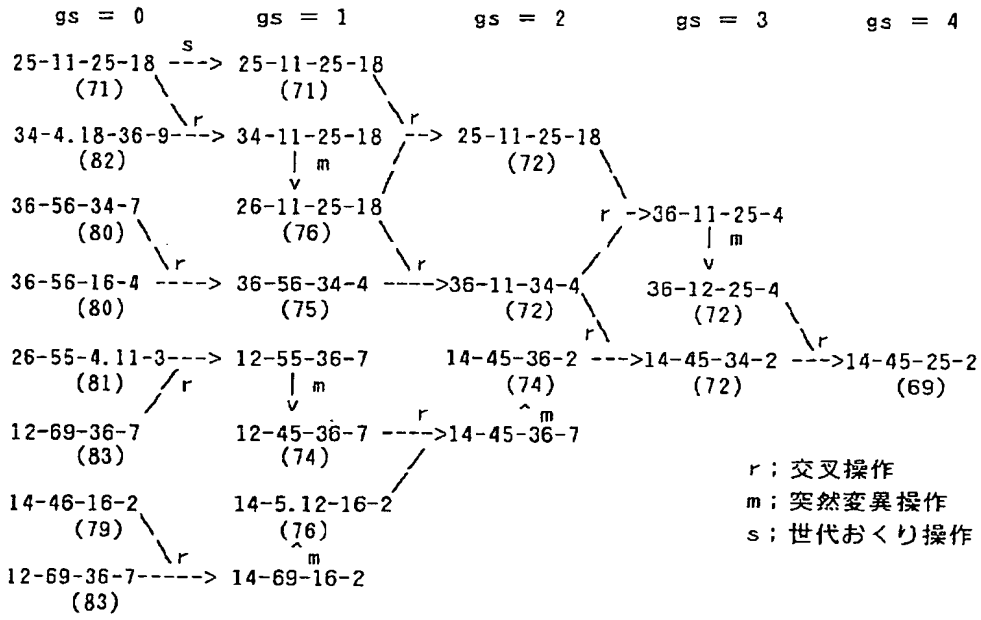
NPOPL = 20

gs 0	min	71	max	118	avr	85
1	min	69	max	80	avr	73
2	min	69	max	81	avr	71
3	min	69	max	73	avr	70
4	min	69	max	75	avr	70
5	min	69	max	75	avr	69
6	min	69	max	74	avr	69
7	min	69	max	72	avr	69
8	min	69	max	74	avr	69
9	min	69	max	72	avr	69
10	min	69	max	72	avr	69

【図１３】

図　１３

（ａ）局所探索なし



（ｂ）局所探索あり